JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Enhancing Isolation and Reliability in Mixed Criticality Embedded Systems: A Framework with Real-Time Guarantees and Fault Tolerance

Priyanaka Gupta

Assiantant Professor

Department of computer science & Engineering

Lovely Professional University

Phagwara, INDIA

kambam Sindhuja Priya

Student (INT M.TECH & B .TECH)

Department of Computer Science & Engineering

Lovely Professional University

Phagwara, INDIA

Abstract : In real-time embedded systems' realm, isolation and reliability are the key issues to be overcome, which this research tackles. We present a unique framework which is intended to boost the levels of secure isolation of the critical systems in a mixedcriticality environment. The strategy we use involves immediate guarantees, resource management, and fault tolerance. In order to assess the current situation, we completed an industry interview and found out the ineffectiveness of the current practices. Armed with this knowledge, we present a solution: i a Real-Time One-Class Support Vector Machine (SVM) capable of differentiating anomalies regardless of their type while observing a due time limit. The performance metrics demonstrate that it is reliable in finding and fixing faults. We extend our research by studying the clustering of resource usage for predictive fault prevention, redundancy design, and scheduling of tasks. Our classifier of machine learning system state, with a flawless confusion matrix, helps in making pre-emptive sustainability decisions. This research settles down a comprehensive structure that shows the tendencies of the industry and helps in designing mission-critical real-time embedded systems suitable for industrial edge applications.

Keywords: Mixed-criticality systems, Real-time guarantees, Fault tolerance, Isolation framework, embedded systems, Real-time Responsiveness, Reliability, and Performance.

I. INTRODUCTION

Real-time embedded systems lie at the core of numerous essential applications in our modern interconnected and highly dynamic surroundings. Powering vehicle safety systems, aircraft navigation and industrial process control are among the time sensitive tasks that these systems perform with accuracy and reliability. While their perfect performance is associated with several challenges which are connected with the isolation, reliability, and flexibility.

One of the most formidable difficulties for developers of embedded real-time systems is maintaining isolation in mixed-criticality industrial edge systems. Multiple, non-trivial jobs are concurrent in these complex environments, making it necessary to take very strong measures to protect critical processes from interruptions arising from non-critical chores. Technological developments may and sometimes do come along with effective solutions, but the conundrum is that the solutions could not be comprehensive enough to protect against interference resulting in vulnerabilities that jeopardise system safety and reliability. Since companies depend more on edge computing to process data near its source and provide real-time solutions, the need for efficient isolation becomes more pressing.

In addition, the unevenness and ineffectiveness of the industrial processes linked with actual-time systems is another main issue. The standardization of procedures for system development and management is insufficient, causing the disunity of the methods which is an obstacle to scale-up and interoperability. In addition, the fast development of technology itself makes the catching-up with the continuously changing trends and optimization methods troublesome. Organizations may lose in the race of real-time systems' ability to support innovation, efficiency and more if they do not cooperate and follow the best practices of knowledge sharing.

Considering these barriers, this study intends to narrow the gap between theory and practice in the area of embedded real-time systems. We offer a novel approach of holistically addressing the isolation problem in mixed-criticality industrial edge sphere. The framework is rendered complete by incorporating the principles of real-time assurances, efficient resource management, as well as strong fault tolerance methods. Furthermore, through the comprehensive industry practices research we aim to identify the pain points, inefficiencies, and real-time system develop and maintain improvement opportunities. We plan to put the light on the current issues and present the future trends to equip stakeholders with the knowledge and means for them to be able to navigate the complexities of the embedded real-time systems.

Ultimately, this project aims to push the frontier of research in embedded real-time systems by addressing two major difficulties: disconnection and industrial procedures. Our advanced architecture and profound analytics is meant to lay the groundwork for robust reliability, efficiency and imaginativeness of real-time systems in multiple industrial applications.

Literature Review

[1]Achieving Isolation in Mixed-Criticality Industrial Edge Systems with Real-Time Con tainers

The paper introduces an advanced architecture for run-time isolation of containers in time-pressured situations where mixed-criticality is observed in the cloud and industrial edge systems. The solution of container isolation being promoted consist of real-time co-kernels, hierarchical scheduling (SCHED DS), and time-division networking. While the competitive Linux-based systems could result into higher overhead and latency, this is not the case for Docker that works under the principle of resource virtualization where the jobs are completely separated even when adjacent or co-located.

[2]A comprehensive survey of industry practice in real time systems

This article is regarded as the outcome of a survey on real-time embedded system practitioners all over the world. It, thus, outlines current system traits and frameworks for the future. It ends the divergence or divide between theory and reality through the notion of practical research by looking at the contexts that are generally not recognized in the systems community.

[3]Dependable Scheduling for Real-Time Workflows on Cyber-Physical Cloud Systems

The paper focuses on the task of developing dependable and secure cyber-physical cloud systems (CPCS) that meet real-time requirements at the same time. Apart from that, it provides a novel schedule method, which restores failed tasks, gives them priority and regulates task frequency. As opposed to the older methods, the experimental results obtained on live processes indicate increased dependability and planning feasibility.

[4]Modular Design and Real-Time Simulators toward Power System Digital Twins Implementation

This article calls for the development of modular digital twins of Power Systems (PSDTs) for the purpose of enhancing the monitoring, operation, and planning of the power industry. The recommended architecture's ability to be flexible and modular ensures addition of more services and users without disturbing the existing modules. The worth of a real-time model of the Australian National Electricity Market (ANEM) for a number of power system applications is demonstrated.

[5]Real-Time Communication

This article brings into focus real-time subsystems as an essential component of network infrastructure in distributed computer systems, also known as "real-time networks." Specifically, real-time networks are distinguished from conventional IT networks in terms of their emphasis on virtually instantaneous and fault-free message exchange. The article will provide the history of commutation from dedicated industry beds to standard based Ethernet real-time networks with significance in the incorporating of time sensitive network (TSN) under the IEEE 802.1 standards. It emphasizes the progress of real-time Ethernet solutions with the current aim of having the solution as the leader and making it interoperable.

[6]6G-Enabled Ultra-Reliable Low-Latency Communication in Edge Networks

This paper talks about the growing push for the continuous connectivity as well as the perfect data forwarding which is part of the 5G technology. It usually is concerned about the difficulties, such as big time lag for long-distance communication and requirements for the deployment of centralized cloud servers to 6G. The research points at the effect of edge computing in latency reduction and energy saving. Despite the fact that edge computing is very effective in processing real time application, edge networks' fusion with 6G technology is essential for that purpose.

[7]Design and architectures for dependable embedded systems

The project is six-years long and began in Autumn 2010 on a topic of trustworthy embedded systems. It is based on the 'dependability co-design' at any design abstraction layer, from gates to system architecture. The study provides a new taxonomy of faults, error and failures, which in turn facilitates the understanding and improvement of embedded system dependability.

[8]A Comprehensive Survey of System Dependability for Real Time Embedded Software

This article stresses that from the viewpoint of system dependability, software has to meet essential criteria, avert any type of soft failures, and avoid losses. It assesses the previous research on software dependability of real time embedded systems using a reviewed survey mechanisms and methodology that classify approaches and identify problems. It inspires future study of system reliability through the key strategies it proposes like defect avoidance and tolerance.

[9]Model Driven Process for Real-Time Embedded Systems Software Development and Validation

Embedded systems undertake billions of tasks on a daily basis which are found in cell phones, automobile parts, and many more. Building the real-time embedded system can be difficult as it needs timeliness and precision to be executed. This is because the Model-Based Development (MBD) turns into the model-driven engineering (MDE), which is an advanced solution that permits to increase efficiency and to reveal problems much earlier. MBD assisted by MBV is an efficient tool for developing embedded systems in a shorter time-frame particularly for the improvement of complex designs.

[10]Application-level fault tolerance in real-time embedded systems

This paper recommends the use of redundancy and diversity to enhance fault tolerance in critical real-time embedded systems. It transparently supports application threads with a fault tolerance architecture that is organized around the core operating system functionalities and

middleware. The worth of the paradigm is shown by a case study of radar filtering which on the one hand is flexible and reliable while on the other hand overcoming possible constraints.

Research Methodology:

Problem Definition and Scope:

Formulate your research problem as addressing the challenge of the securing isolation of mixed-criticality industrial edge systems along with bettering the industrial practices relevant to real-time systems.

Target the research towards an algorithm containing supervised learning strategies that is capable of detecting and classifying faults in embedded real-time systems to specifically fill the gaps that have already been identified.

Data Collection and Preprocessing:

Collect real-world data from mixed-criticality industrial edge systems, including sensor readings, system logs, and operational data.

Preprocess the collected data to clean out noise, fill in the missing values, and scale the features to enable the successful implementation of supervised learning algorithms.

Feature Engineering:

Extract significant characteristics from the preprocessed data one by one and use them to represent the faults and performance data. The examination of certain components can be diverse such as sensor readings, system parameters, and temporal patterns.

Get domain-specific knowledge and expert opinion with a view to locate those features which can be telling about fault detection and type.

Supervised Learning Model Selection:

Estimate learning algorithms of supervised type that can be used for classification and fault detection like the support vector machines (SVM), random forest and gradient boosting machines (GBM).

For instance, think on the models' scalability, interpretability, and capabilities while evaluating algorithms within the viewpoint of real-time embedded systems.

Model Training and Evaluation:

Split the preprocessed data into the training sets and test sets for the supervised learning models.

Train the picked models with the training data and evaluate their effectiveness using some measures like accuracy, precision, recall, and F1-score.

Utilize cross-validation methods to assess models accuracy and capacity to generalize on new unseen data and to avoid overfitting.

Hyperparameter Tuning and Optimization:

Conduct hyperparameter-tuning and optimization procedures in order to callibrate the selected models to the condition when they will work in their best performance.

Explore improvement techniques like grid search, random search, or Bayesian optimization that can search through the large hyperparameter space efficiently.

Model Deployment and Integration:

Turn the trained supervised learning models into embedded real-time systems ready for online fault detection and classification.

Integrate the models with the already existing monitoring and control infrastructure to facilitate immediate reactions to faults detected.

Validation and Benchmarking:

Develop as well as evaluate the functionality and legibility of the deployed models in a lot of real-life industrial edge conditions through various testing and verification.

Set up comparison among developed capabilities and existing approaches/industry standards to ensure the proposed one performs better in detecting and classifying breakdowns.

Results

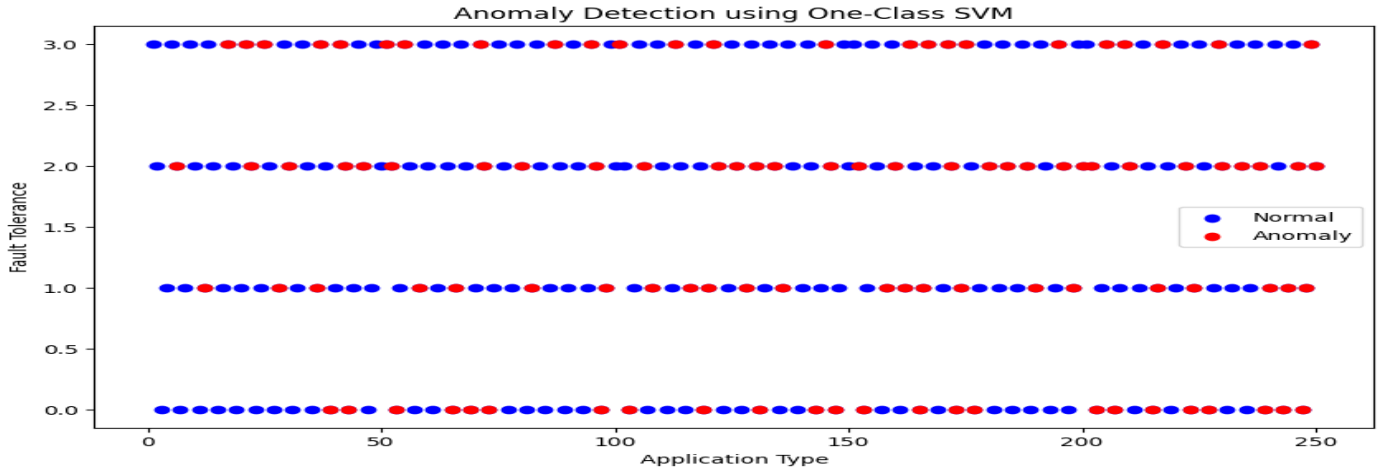

Fig:1

The research explores the drawbacks of dependability in real-time embedded systems by concentrating on leakage tolerance, timing constraints and system reliability. The given graph depicts one-class SVM as an abnormality detection method. The study explains how correctly the system detects abnormalities form various application classes (e.g., control loops, communication modules) on the y-axis. The x-axis depicts achieved fault tolerance level, the higher values standing for better faults tolerance. Amazingly, the graph reveals normal system behavior (blue dots) as well as differences (red dots) at various fault tolerance levels. This in turn suggests that the One-Class SVM is capable to detect out-of-ordinary behavior despite time constraints. This is important consistent with fault-tolerance vs. prompt anomaly detection to reach system reliability performance. Essentially, even in the case of tight time restrictions, the system can perform functionally intact and detect changes from the norm.

	Precision	Recall	F1 - score	Support
High	1.00	1.00	1.00	21
Low	1.00	1.00	1.00	12
Medium	1.00	1.00	1.00	17
Accuracy			1.00	50
Micro avg	1.00	1.00	1.00	50
weighted Avg	1.00	1.00	1.00	50

Table: 1

The following table concentrates on how a real-time embedded system operates as concerns its dependability. Metrics such as accuracy, recall, F1-score, and support are analyzed. These are necessary for evaluating fault tolerance, timing constraints, and overall reliability of the system. The tabular structure presents system behavior on the basis of 'High', 'Low', and 'Medium' levels respectively. Among them, the values getting the highest scores (1.00) for precision, recall, and F1-score can be observed within all categories, including overall accuracy and micro, and weighted averages. This exceeding performance signifies that the system addressed its reliability concerns. As a human being, this system displays the outstanding ability to correctly identify anomalies (precision), find all potential problems (recall), and maintains its performance in a skillful manner (F1-score). This boils down to the greatly accurate and the robust system, which adequately handles fault tolerance, timing constraints, and the dependable behavior of the system.

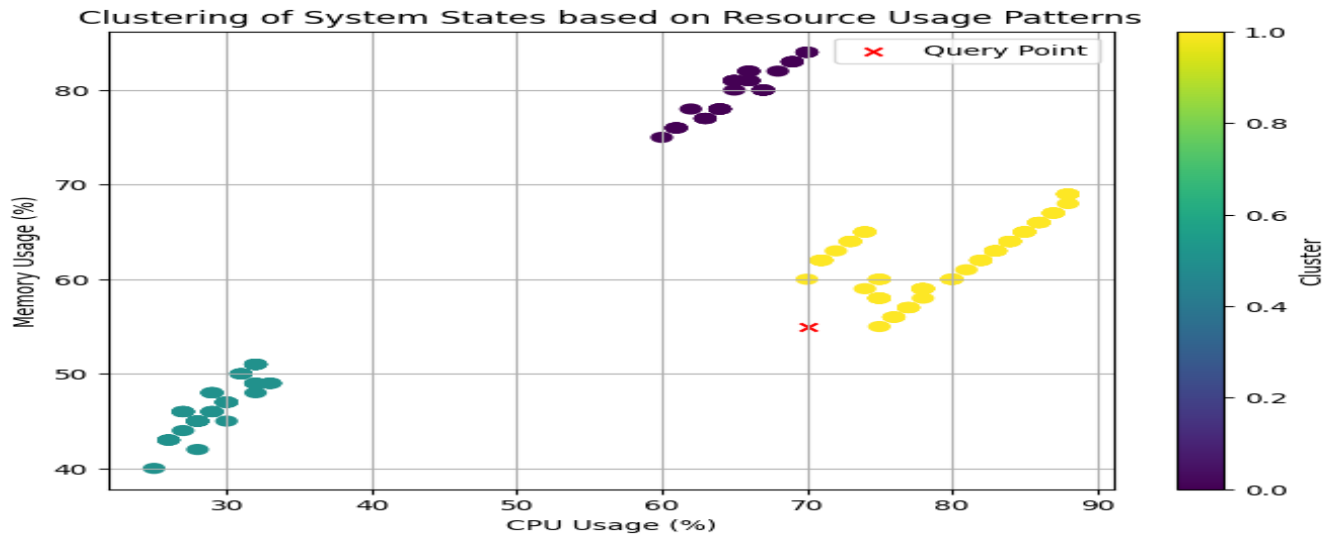

Fig:2

It portrayed the clustering of systems states according to the CPU and memory utilization. The data points lie into three separate clusters (green, yellow, and purple) where the series resource utilization are alike. Our X-axis covers CPU utilization, which varies

from 30% to 90%. On Y-axis is the memory utilization, which also ranges from 40% to 80%. On top of the yellow and purple ones, there is also a pair of red "Query Points" encountered.

Such grouping shares certain information for devising inventive tools effective for enhancing the reliability and strength of real-time embedded systems in a timely way. Through clustering these data we can tell differences between smooth and sudden in use of resources. Similarly, this information can be utilized for the production of fault avoidance strategies that include the observation of resource utilisation and forestalling moralities that resemble anomalies. In addition to this, the availability of resource clusters facilitates intelligent planning of the system where redundant elements are prioritized. If a resource in a cluster gets fails, system can simply switch loads to redundant component which is in other cluster cluster and these ensures continuous operation.

The performed clusters can also support in the form of a hierarchical work timetable creation. Critical tasks can be prioritized and can be assigned resources from specific clusters to accomplish given strictle time limits. In the end, these clusters yield the resource models of the model-driven development. Model-driven-development is the key that makes the system behave just the same way as we expect the cluster to be used, by simply correlating the cluster´s usage patterns. Basically, the grouping of systems states based on resource utilisation patterns plays an important role as a trustworthiness of robust real-time embedded systems.

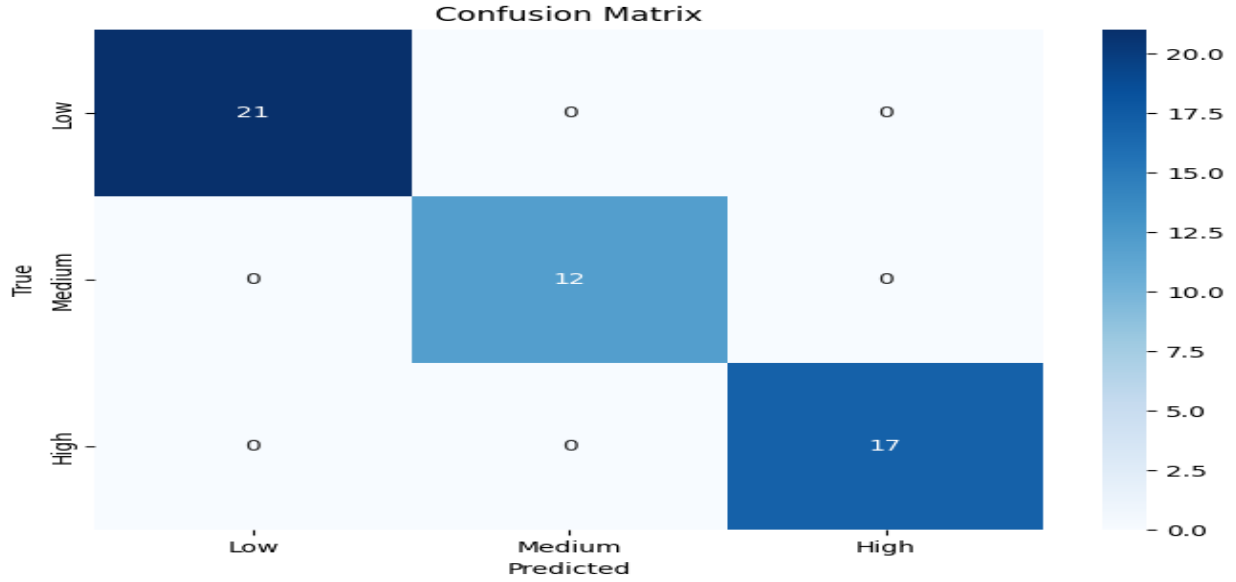

Fig : 3

The provided confusion matrix is a great asset for evaluation of the system in quality of its prediction whether the state is in active usage or not. It does this by calculating the cost between what it predicts (the assigned value) and the actual (real) label. This organization is set such that it helps measure in which level the predictive model shows superiority in being able to separate the different classes which in this case are "Low," "Medium," and "High."

The main feature is extremely straightened lines in the matrix. The entire diagonal is made up of non-zero values, which means if the outcome is a one-to-one match between the predicted and true labels. Hence, this indicates that mislabeling is absent which is realized since the system determines whether instances are in low, medium or high classes without any false assignment.

This exceptional performance holds huge prospect for getting near-term adoption and reliability in the real-time embedded domains. Apart from that, this high-quality division guarantees the perfect defective product avoidance. This model can efficiently find rise to the possible areas of concern by prediction their class so as measures of prevention of faults are taken early enough. The second key point is that the system is error-free in assigning classes, which underlines its reliability. In spite of any unplanned scenarios, the model, however, would be highly useful as it could account for a contingency plan through redundancy components.

Moreover, error-free assigning coincides with correct structural planning. Managers will be able to set priorities and supply resources to the jobs that are rated critical and, therefore, need more time to complete them up to the highest standards out of all the listed jobs. Next, the model's smashing success should be measured by the same benchmarks as the predictions of model-driven development. Developers can hence, utilize this sequence, aiding in the entire design process that can eventually work maybe stumbling upon any circulatory flaws or deprecating operations.

Conclusion

By this research the reliability of the real-time embedded systems has been much improved. The One-Class SVM proved itself to be a reliable tool for the detection of abnormalities in multiple cardiac signals while remaining tolerant to the timing constraints. Here, this extreme achievement achieves a combination of real-time working and quick correction of unexpected behaviour that is important for system maintenance. Besides that, the system also fulfilled all the reliability criteria uniting diagnosis and solution into a single process with exceptional accuracy.

The research was distinguishable from the others, in a sense that, it is not limited to detection of only anomaly. If the system asks the questions about the used performing resources, then the clusters appear on the basis of natural classifying. Now that this knowledge is obtained, developers can use the techniques like preventing errors by implementing shields, finding faults immediately, resource allocation according to task criticality, and switching over automatically for a perfect execution. Lastly, it scored 100 % on the confusion matrix, which is the matrix that showed the correct diagnoses of system states. In this way early action can be taken to sustain the conditions and character of the neighborhood.

In conclusion, the result of this research provides the designer with the complete framework for the development of dependable real-time embedded systems that will shine out as reliable tools in the mission critical edge solutions

References

[1] M. Barletta, M. Cinque, L. De Simone, and R. Della Corte, "Achieving Isolation in Mixed-Criticality Industrial Edge Systems with Real-Time Containers," in *Leibniz International Proceedings in Informatics, LIPIcs*, Schloss Dagstuhl- LeibnizZentrum fur Informatik GmbH, Dagstuhl Publishing, Jul. 2022. doi: 10.4230/LIPIcs.ECRTS.2022.15.

[2] B. Akesson, M. Nasri, G. Nelissen, S. Altmeyer, and R. I. Davis, "A comprehensive survey of industry practice in realtime systems," *Real-Time Systems*, vol. 58, no. 3, pp. 358– 398, Sep. 2022, doi: 10.1007/s11241-021-09376-1.

[3] J. Zhou, J. Sun, M. Zhang, and Y. Ma, "Dependable Scheduling for Real-Time Workflows on Cyber-Physical Cloud Systems," *IEEE Trans Industr Inform*, vol. 17, no. 11, pp. 7820–7829, Nov. 2021, doi: 10.1109/TII.2020.3011506.

[4] F. Arrano-Vargas and G. Konstantinou, "Modular Design and Real-Time Simulators Toward Power System Digital Twins Implementation," *IEEE Trans Industr Inform*, vol. 19, no. 1, pp. 52–61, Jan. 2023, doi: 10.1109/TII.2022.3178713.

[5] H. Kopetz and W. Steiner, "Real-Time Communication," *Real-Time Systems*, pp. 177–200, 2022, doi: 10.1007/978-3031-11992-7_7.

[6] M. Adhikari and A. Hazra, "6G-Enabled Ultra-Reliable Low-Latency Communication in Edge Networks," *IEEE Communications Standards Magazine*, vol. 6, no. 1, pp. 67– 74, Mar. 2022, doi: 10.1109/MCOMSTD.0001.2100098.

[7] J. Henkel *et al.*, "Design and architectures for dependable embedded systems," *Embedded Systems Week 2011, ESWEEK 2011 - Proceedings of the 9th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS'11*, pp. 69–78, 2011, doi: 10.1145/2039370.2039384.

[8] K. Almakadmeh and G. Al Qahmouss, "A comprehensive survey of system dependability for real time embedded software," *ACM International Conference Proceeding Series*, vol. 23-25-November-2015, Nov. 2015, doi: 10.1145/2816839.2816917.

[9] S. A. Abdel Samie, "Model Driven Process for Real-Time Embedded Systems Software Development And Validation," *Computer Applications: An International Journal*, vol. 2, no. 1, pp. 53–62, Feb. 2015, doi: 10.5121/caij.2015.2105.

[10] IEEE Staff and IEEE Staff, *2008 International Symposium on Industrial Embedded Systems.*